Pre-computation Based Heuristic for Inter-Domain QoS Routing

Ahmed Frikha, Samer Lahoud

Pre-computation Based Heuristic for Inter-Domain QoS Routing

Ahmed Frikha and Samer Lahoud
University of Rennes 1, IRISA, 35042 Rennes Cedex, France
Email:{afrikha,slahoud}@irisa.fr

***Abstract*—In the present paper, we propose a novel fast heuristic for inter-domain QoS routing, named ID-PPPA. ID-PPPA is based on a pre-computation scheme. The pre-computation scheme attempts to solve the QoS routing problem while keeping a low response time by computing in advance a set of QoS paths. Our solution preserves domain confidentiality and solves the scaling problem related to inter-domain routing by distributing computations over the domains. Theoretical analysis proves that the ID-PPPA algorithm has a low computational complexity, which is very necessary for a pre-computation algorithm to deal with dynamic changes of the network link state information. Moreover, extensive simulations confirm the efficiency of our algorithm in terms of success rate and quality of the computed path.**

I. Introduction

Quality of Service (QoS) routing, also known as multi-constraint routing, is one of the primary mechanisms for providing guaranteed performance to IP-based applications such as Video on Demand (VoD), and Voice over IP (VoIP). It consists in the computation of a path which satisfies multiple QoS constraints such as the cost, the delay, etc. Two schemes for path computation are proposed in the literature. First, the on-demand scheme that consists in the computation of a path which satisfies the QoS constraints upon the reception of a QoS request. Second, the pre-computation scheme that consists in preparing in advance a set of paths satisfying predetermined QoS requests. Then, at the reception of a QoS request, it attempts to rapidly provide a feasible path among the pre-computed paths. Unlike the on-demand computation, the pre-computation allows to speed up the response time while solving the QoS routing problem..

Extending the QoS routing to an inter-domain level faces essentially two challenges: confidentiality and scalability. First, information about internal topology or available resources in the network is confidential, as the operators can be in competition. Hence, complete routing information throughout a multi-domain network is unvailable. Second, computing the path which satisfies the QoS constraints throughout a sequence of domains is more complex because of the large number of nodes involved in the computation. Currently, the inter-domain routing protocol is BGP. This protocol does not take into account QoS constraints. Many extensions for BGP are proposed to support QoS routing [1]-[2]. However, the QoS capabilities of these propositions remain limited [3]. The research community has recently been exploring the use of distributed architectures, such as the PCE (Path Computation Element) architecture [4], to solve the inter-domain QoS routing problem. Distributing the computation over domains preserves the confidentiality across domains and reduces the response time. Therefore, our proposed solution for inter-domain QoS routing relies on the use of a distributed architecture, such as the PCE architecture. Precisely, we propose a novel fast and distributed heuristic for the inter-domain QoS routing based on pre-computation, named ID-PPPA (Inter-Domain Primary Path-based Pre-computation Algorithm). ID-PPPA has a low theoretical computational complexity. Furthermore, it can reduce the time to establish the connection since it is based on pre-computation. It preserves the domain confidentiality clauses and overcomes the scaling problem since it relies on a distributed architecture. We also strengthen the analysis of our novel algorithm by performing exhaustive simulations.

II. Problem Formulation

Before formally defining the problem, we first introduce some notations. Let $G(N, E)$ denote a network topology graph, where N is the set of nodes (routers) and E the set of links. Let m be the number of constraints. In our study, we consider additive constraints. An m-dimensional weight vector is associated with each link $e \in E$. This vector consists of m non-negative QoS weights $w_i(e)$, $i = 1..m$. Let p be a path in the graph $G(N, E)$ and $w_i(p)$ be the weight of p corresponding to the additive metric i. Thus, $w_i(p)$ is given by the sum of the i^{th} weights of its component links: $w_i(p) = \sum_{e_j \in p}(w_i(e_j))$. Let $\overrightarrow{W}(p) = (w_1(p), w_2(p), ..., w_m(p))$ denote the weight vector of the path p.

Definition 1: The MCP problem. Given a source node s and a destination node d and a set of constraints given by the constraint vector $\overrightarrow{C} = (c_1, c_2, .., c_m)$, the Multi-Constraint Path (MCP) computation problem consists of finding a path p which satisfies $w_i(p) \leq c_i$, $\forall i \in 1..m$. Such a path p is called a feasible path. The MCP problem is $\mathcal{NP}$ complete [5], consequently the computational time for solving this problem is high.

A feasible path p is called non-dominated if there does not exist a path p' which satisfies: (1) $w_i(p') \leq w_i(p), \forall i \in 1..m$ and (2) $\exists j \in 1..m$ for which $w_j(p') \prec w_j(p)$. To speed up the computation, dominated paths can be discarded from the computation search space of the algorithms without affecting neither the success rate of the algorithm nor the quality of the computed path, according to [6].

III. The ID-PPPA algorithm

In this section, we present our novel inter-domain multi-constraint pre-computation algorithm. ID-PPPA consists of two phases: an offline phase and an online phase. In the offline phase, named per-domain pre-computation, ID-PPPA pre-computes a set of QoS paths in each domain satisfying a set of pre-determined QoS request. In the second phase, named inter-domain path computation, ID-PPPA attempts to compute an end-to-end path by combining the paths pre-computed in each domain.

A. The offline phase

In this phase, each domain pre-computes a set of internal QoS paths. These paths will be used later at the online phase to compute an end-to-end path.

In practice, some QoS metrics are more critical for certain applications, such as the delay for the VoIP-based applications. Therefore, we pre-compute for each QoS metric the path which minimizes the weight corresponding to this metric. For example, it pre-computes the path which minimizes the delay; this path can be useful for the VoIP-based applications. Let us consider a domain D. ID-PPPA pre-computes a set of paths from each border node of the domain D toward the other node of D as well as the entry border nodes of the neighbor domains. Let m be the number of the QoS metrics, n_1 be a border node of D, and n_2 be a node of D or an entry border node of a neighbor domain, ID-PPPA computes m shortest paths from n_1 to n_2. Each shortest path minimizes a single QoS metric and is called primary path. Hence, from each border node of D, ID-PPPA computes m shortest path trees. Each shortest path tree, also called a primary path tree, is computed using the Dijkstra algorithm and considering a single weight component w_i, where $i \in 1..m$. Therefore, ID-PPPA exectutes the Dijkstra algorithm m times per border node. Consequently, the complexity of the offline phase of ID-PPPA is then given by: $O(B * m(N \log(N) + E))$, where B is the number of border nodes of the domain. After computing these paths, ID-PPPA filters all dominated paths. The remaining paths are stored in a database to be used in the online phase.

After an eventual change in the network state, the pre-computed paths may be not valid. Therefore, executing the offline phase of ID-PPPA periodically or using a network state-dependent threshold is required. The low theoritical complexity of this phase enables to cope with a dynamic change in the network state information since it allows domains to rapidly pre-compute new valid paths. Setting a period or a threshold for the execution of the offline phase depends on the variability of the traffic in the network. This issue goes beyond the problem considered in this work.

B. The online phase

This phase is triggered upon the reception of a QoS request. Our algorithm attempts to compute an end-to-end path by combining the pre-computed paths in each domain. When receiving a QoS request, the service provider computes the best domain sequence that links the source and the destination domain according to the cooperation policy [4]. The end-to-end path computation starts at the destination domain toward the source domain following the selected domain sequence. Note that, without loss of generality, we rely on backward computation according to the PCE architecture.

First, the destination domain selects the pre-computed paths that lead to the up-stream domain following to the selected domain sequence. Then, these paths will be sent to the up-stream domain using a novel compact structure named VSPH (Virtual Shortest Path Hierarchy[1]). This structure contains only the end nodes of the paths (the destination node and the entry border nodes of the up-stream domain) and the weight vector of each path. This structure allows the confidentiality of the domains to be preserved. When receiving a VSPH, an intermediate domain combines the paths in the VSPH with the internally pre-computed paths. Then, it selects the feasible paths by comparison with the constraint vector, computes a new VSPH and sends it to the up-stream domain. Finally, the computation is stopped when a feasible path linking the source and destination is found, or when no feasible path is found. In this case, the QoS request is rejected.

Let D denotes an intermediate domain, n be an entry border node of D, and B_{max} denote the maximum number of border nodes between two domains. There are at most m paths in the VSPH from the destination to n and at most mB_{max} pre-computed paths to reach the upstream domain from n. Hence, the complexity of combining the pre-computed paths with the received paths at the level of the entry border node n is given by $O(m^2 B_{max})$. Considering all entry border nodes of D, the global complexity of this phase is given by $O(m^2 B_{max}^2)$.

IV. Simulation and analysis

In this section, we evaluate the performance of our novel algorithm ID-PPPA by comparison with the exact pre-computation algorithm pID-MCP [8]. The exact algorithm pID-MCP finds a feasible path, whenever such a path exists. However, the drawback of this algorithm is its high computational complexity. The simulations are performed using the realistic topology SYM-CORE [9]. We associate with each link three additive weights generated independently following a uniform distribution [10, 1023]. The QoS constraints are also randomly generated in the constraint generation space Z shown in figure 1. This space is deduced by computing p_1, p_2, and p_3 the three shortest paths which minimize the first, the second, and the third metric, respectively. The problem is not $\mathcal{NP}$-Hard outside Z, i.e. either infeasible or trivial. As shown in figure 1, we select ten zones Z_i, $i = 1..10$ from this space and we browse them from the strictest constraint zone Z_1 to the loosest constraint zone Z_{10}. Then, we assess the performance of our algorithm according to these zones.

In the following, we compare our algorithm ID-PPPA with the exact algorithm pID-MCP according to two performance

[1]The hierarchy is a structure which enables the storage of multiple paths between any two nodes [7].

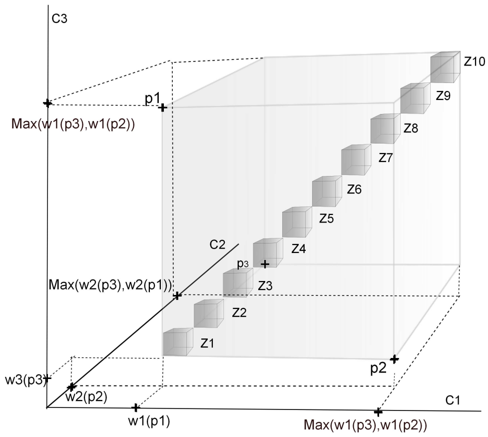

Fig. 1. Constraint generation zones for $m = 3$

criteria. First, the Success Rate (SR) given by the percentage of the requests for which a feasible path is found. Second, the Cost (C) given by the value of the path length function $c(p) = 100 * \max_{i \in 1..m}(\frac{w_i(p)}{c_i})$ for the best path computed. The cost (C) is considered only for the request for which ID-PPPA finds a feasible path. This performance criteria indicates the quality of the computed path since it gives a measure of the distance between the path weights and the constraints.

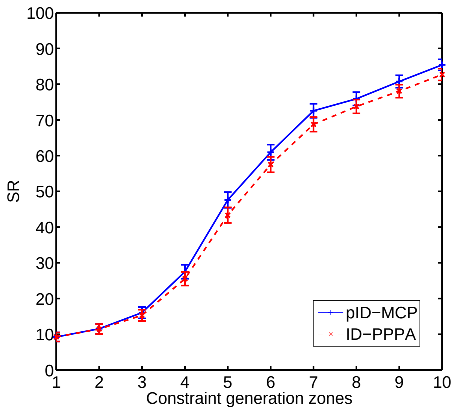

Fig. 2. Success rate according to the constraint generation zones for $m = 3$

Figure 2 shows the success rate of ID-PPPA and pID-MCP according to the strictness of the QoS constraints. The success rate of the algorithms increases when the constraints become less strict. In fact, the probability to find a feasible path is higher when the constraints are looser. The fundamental result deduced from figure 2 is that the success rate of our novel heuristic ID-PPPA is very close to that of the exact algorithm pID-MCP, which is advantageous considering that its complexity is reduced.

Figure 3 shows the cost C of the best paths computed by our heuristic ID-PPPA and the exact algorithm pID-MCP according to the strictness of the QoS constraints. The cost C of the algorithms decreases when the constraints are less strict. In such cases, the quality of the best computed path becomes better for two reasons. First, the number of feasible paths is higher when constraints are not strict. Hence, the probability to find a path with better quality increases. Second,

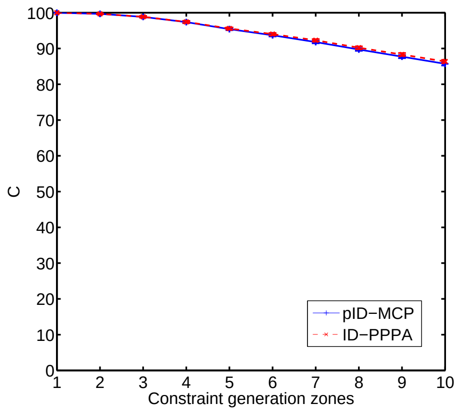

Fig. 3. The cost C according to the constraint generation zones for $m = 3$

when constraints are looser, the values of the weight vector components of the best path are far below the values of the constraint vector components. The most relevant result of this figure is that the cost C of the path computed by ID-PPPA is slightly higher than that of the one computed by the exact algorithm. This proves the efficiency of our algorithm in term of computed path quality.

V. Conclusion

In this paper, we presented a new fast inter-domain path computation heuristic based on primary path pre-computation, called ID-PPPA. Our heuristic relies on a distributed architecture, such as the PCE architecture, which allows domain confidentiality to be preserved. Furthermore, ID-PPPA has a fast response time since it is based on a pre-computation scheme. The low computational complexity of the offline phase allows pre-computations to be executed rapidly to deal with eventual changes in the network load. Extensive simulations prove that our heuristic has a high success rate, very close to that of the exact algorithm pID-MCP. Moreover, the quality of the path computed by our heuristic is very close to the optimal path.

References

[1] T. Knoll, *BGP Extended Community Attribute for QoS Marking*, draft-knoll-idr-qos-attribute-02, work in progress, IETF, 2009.

[2] D. Griffin, J. Spencer, J. Griem, M. Boucadair, P. Morand, M. Howarth, N. Wang, G. Pavlou, A. Asgari, and P. Georgatsos, *Interdomain routing through QoS-class planes*, IEEE Commun. Mag., vol. 45, no. 2, pp. 88-95, February 2007.

[3] R. Mahajan, D. Wetherall, and T. Anderson, *Towards coordinated inter-domain traffic engineering*, in SIGCOMM Workshop on Hot Topics in Networking (HotNets). New York, NY, USA: ACM, 2004.

[4] A. Farrel, J.P. Vasseur, J.A Ash, *Path Computation Element (PCE)-Based Architecture*, IETF RFC 4655, August 2006.

[5] Z. Wang and J. Crowcroft, *Quality-of-Service Routing for Supporting Multimedia Applications*, IEEE Journal on Selected Areas in Communications, vol. 14, no. 7, pp. 1228-1234, 1996.

[6] P. Van Mieghem and F.A Kuipers, *Concepts of exact QoS routing algorithms*, IEEE/ACM Transactions on Networking, vol. 12, no. 5, pp. 851-864, October 2004.

[7] M. Molnar, *Hierarchies for Constrained Partial Spanning Problems in Graphs*, Tech. Rep. 1900, IRISA(2008).

[8] G. Bertrand, S. Lahoud, M. Molnar, G. Texier, *Recent Advances in Providing QoS and Reliability in the Future Internet Backbone*, Nova Science, Hauppauge, NY, USA (2010).

[9] S. Dasgupta, J. de Oliveira, J.P. Vasseur, *Performance analysis of inter-domain path computation methodologies*, draft-dasgupta-ccamp-path-comp-analysis-02, work in progress, IETF (2008)